Exploring Discretization Error in Simulation-Based Aerodynamic Databases

Michael J. Aftosmis[1] and **Marian Nemec**[2]

Keywords: keywords: parallel, parametric studies, error, database, adjoint, Cartesian, Cart3D, CFD

ABSTRACT

This work examines the level of discretization error in simulation-based aerodynamic databases and introduces strategies for error control. Simulations are performed using a parallel, multi-level Euler solver on embedded-boundary Cartesian meshes. Discretization errors in user-selected outputs are estimated using the method of adjoint-weighted residuals and we use adaptive mesh refinement to reduce these errors to specified tolerances. Using this framework, we examine the behavior of discretization error throughout a token database computed for a NACA 0012 airfoil consisting of 120 cases. We compare the cost and accuracy of two approaches for aerodynamic database generation. In the first approach, mesh adaptation is used to compute all cases in the database to a prescribed level of accuracy. The second approach conducts all simulations using the same computational mesh without adaptation. We quantitatively assess the error landscape and computational costs in both databases. This investigation highlights sensitivities of the database under a variety of conditions. The presence of transonic shocks or the stiffness in the governing equations near the incompressible limit are shown to dramatically increase discretization error requiring additional mesh resolution to control. Results show that such pathologies lead to error levels that vary by over factor of 40 when using a fixed mesh throughout the database. Alternatively, controlling this sensitivity through mesh adaptation leads to mesh sizes which span two orders of magnitude. We propose strategies to minimize simulation cost in sensitive regions and discuss the role of error-estimation in database quality.

INTRODUCTION

The shift to multi-core CPU architectures has rapidly accelerated the growth of supercomputing resources [1]. This marked increase in the level of high-performance

[1] Michael J. Aftosmis, NASA Ames Research Center, Mail Stop T-27B, Moffett Field, CA 94035, USA

[2] Marian Nemec, ELORET Corp., Mail Stop T-27B, Moffett Field, CA 94035, USA

computing now offers both unprecedented capability and capacity to the aerodynamic simulation community [2]. These systems are capable of aerodynamic simulations with 10^8-10^{10} degrees of freedom, offering ever increasing physical fidelity [3,4]. While such extremely large "capability" simulations are becoming commonplace, the engineering community has focused on the enormous capacity of these systems through an increasing reliance on parametric, trade and optimization studies. In industrial settings, the main role of high-end computing is performance database generation. Aerodynamic databases with 10^3-10^4 simulations have become common and "production CFD" is the mainstay of the workload on this hardware.

The quality of these databases hinges on the level of the discretization error associated with the simulation outputs. Simply put, how much does the mesh influence the results? In this work, we use the method of adjoint-weighed residuals to estimate the discretization error in each simulation and examine the resulting error distribution. Adaptive mesh refinement is used for error control in each case of the database. These investigations use the parallel multi-level Cartesian Euler solver developed in references [5] and [6] to produce the aerodynamic data. This simulation package has been used extensively on large shared and distributed memory systems and has very good parallel scalability [2,3,7]. The robustness and automation of this simulation package has led to its wide adoption for producing aerodynamic databases in support of engineering analysis and design [8,9,10,11].

The use of adjoint-based error-estimates and adaptive mesh refinement in database construction increases the computational cost of each data point because several flow and adjoint solutions are required to generate a mesh that satisfies the user-specified output tolerances. The adjoint-solver, sensitivity calculation, and error-estimator all use the same parallel, multilevel framework as the base Euler solver and all achieve the same high level of parallel efficiency [12,13,14].

In addition to adaptively meshing a simulation, the error-estimation module can also be used in a single-pass mode to assess the error in a specific functional on a given input mesh. Provided that the Euler simulation on this mesh is sufficiently good, this method allows us to accurately assess the discretization-error in a specific output on a particular mesh. This approach can be applied to each simulation in an aerodynamic database, yielding an *a-posteriori* estimate of the error landscape for that database.

Our investigations employ both of these techniques to investigate the role of discretization error in simulation-based aerodynamic data. We begin by briefly reviewing the salient features of adjoint-error estimation using a discrete adjoint solver. We then present both fixed-mesh and error-controlled databases for a token geometry in incompressible, transonic and supersonic flow. Our analysis tracks both error and cost in these databases and aims to quantitatively understand the implications of both strategies.

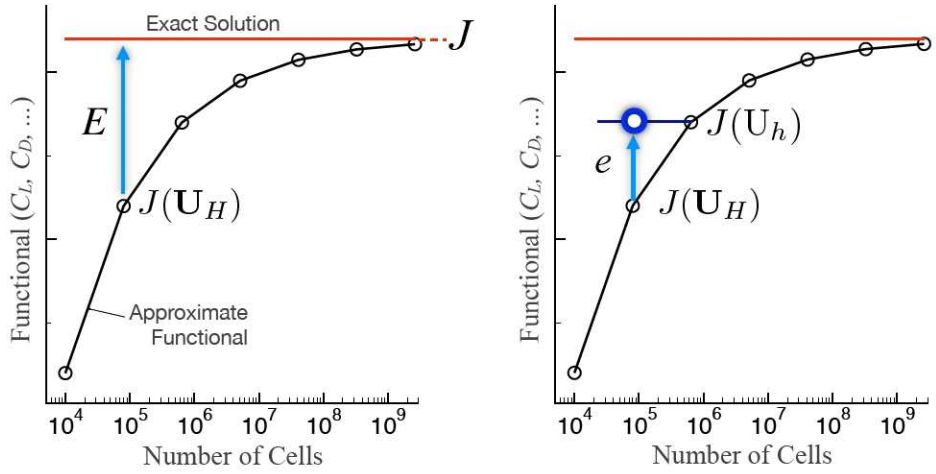

Figure 1: Convergence of functional J with mesh refinement showing the definition of total error, E (left) and relative error, e (right).

ERROR ASSESSMENT AND ESTIMATION

Figure 1 shows two sketches showing convergence of a typical aerodynamic output (C_L, C_D, etc.) with mesh refinement. In the sketch at the left of figure 1, E is the total error in this output due to discretization-error in the numerical solution. This error is defined as the difference between the exact functional value, J, and the value obtained when evaluating this functional using the discrete solution on the current mesh, $J(U_H)$. Rather than attempt to compute E directly, we follow the approach of Ref. [16], and consider instead the simpler problem of estimating how our discrete evaluation $J(U_H)$ would change if we solved on a finer mesh, h. The relative error, e, is sketched at the right of figure 1, and is defined as the difference between the functional evaluations on the current mesh, H, and the finer, embedded mesh h.

$$e = | J(U_H) - J(U_h) | \tag{1}$$

For a second-order method on a sufficiently smooth solution in the asymptotic range, knowing the relative error gives the total error in the output.

$$E = e + \frac{1}{4}e + \frac{1}{4^2}e + \cdots = \frac{4}{3}e \tag{2}$$

Of course, knowledge of the relative error hinges on our ability to evaluate the functional on the fine mesh solution, $J(U_h)$. In [12] and [14] we circumvent this difficulty by approximating the output on the embedded mesh as a function of the coarse mesh flow and adjoint solutions.

$$J(U_h) \approx J(U_h^H) - \underbrace{(\psi_h^H)^{\mathrm{T}}\,\mathbf{R}(U_h^H)}_{\text{Adjoint Correction}} - \underbrace{(\psi_h - \psi_h^H)^{\mathrm{T}}\,\mathbf{R}(U_h^H)}_{\text{Remaining Error}} \tag{3}$$

Where $\mathbf{R}(\,)$ is the spatial operator of the Euler solver, ψ, is the discrete adjoint solution and the notation $(\,)_h^H$ is used to indicate prolongation from the coarser to finer mesh.

References [12] [13] and [15] contain full details of the implementation, and demonstrate the order of convergence of both the adjoint correction and the remaining error terms. In the current work, the remaining error term in eq.(3) is computed by differencing the linear and quadratic prolongations of the adjoint solution. Earlier publications contain detailed verification exercises demonstrating that this formulation achieves its designed mesh-convergence rate for both the adjoint-correction and the remaining-error terms [12, 15].

CONSTANT ERROR DATABASE

As a baseline for investigating the role of discretization error in simulation-based aero-data, we start by computing an aerodynamic database to a fixed error-tolerance. Our prototype is a simple Mach-α sweep over a NACA 0012 airfoil in 2D. In an effort to examine discretization-error and meshing requirements over a wide range of physics, our parametric space covers 12 Mach numbers from low subsonic to moderate supersonic, $M_\infty = \{0.1\text{-}3.0\}$, and 10 different angles-of-attack in the range {0°-12°}. Figure 2 illustrates the extent of this Mach-α space with inset figures (shaded by local Mach number) to illustrate flow in various regimes.

For this study, the objective function was chosen to be a simple unweighted sum of lift and drag on the airfoil.

$$J(U_H) = C_l + C_d \tag{4}$$

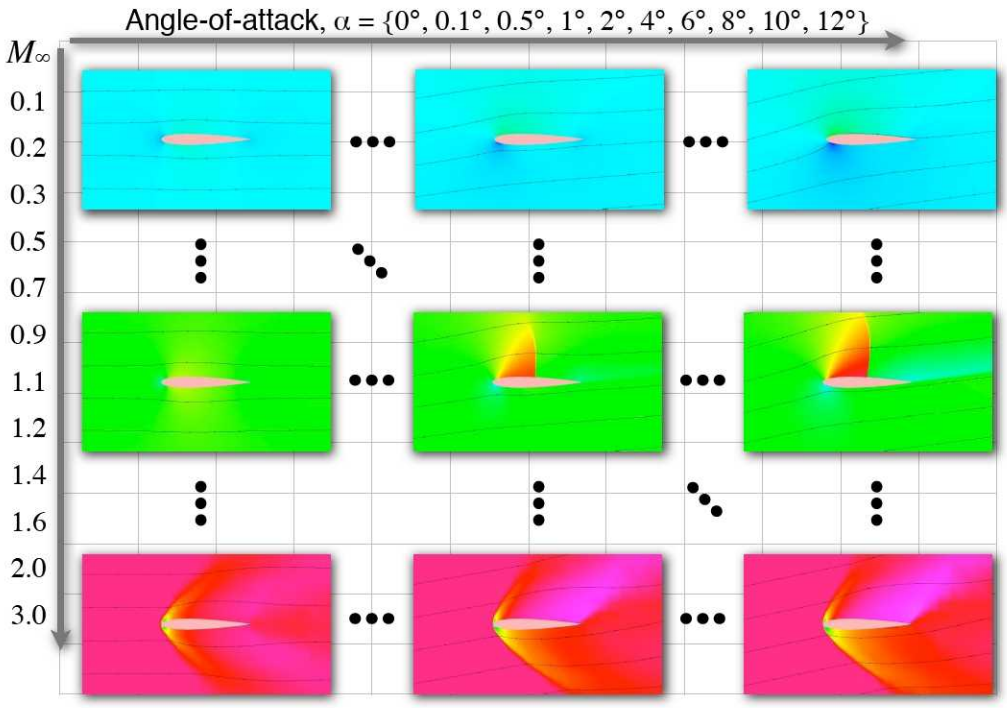

Figure 2: Mach-α wind-space covered by a prototypical aero-database showing examples of flow in various flight regimes. (12 Mach numbers) x (10 incidence angles) = 120 simulations in the database, Mach contours shown for selected cases.

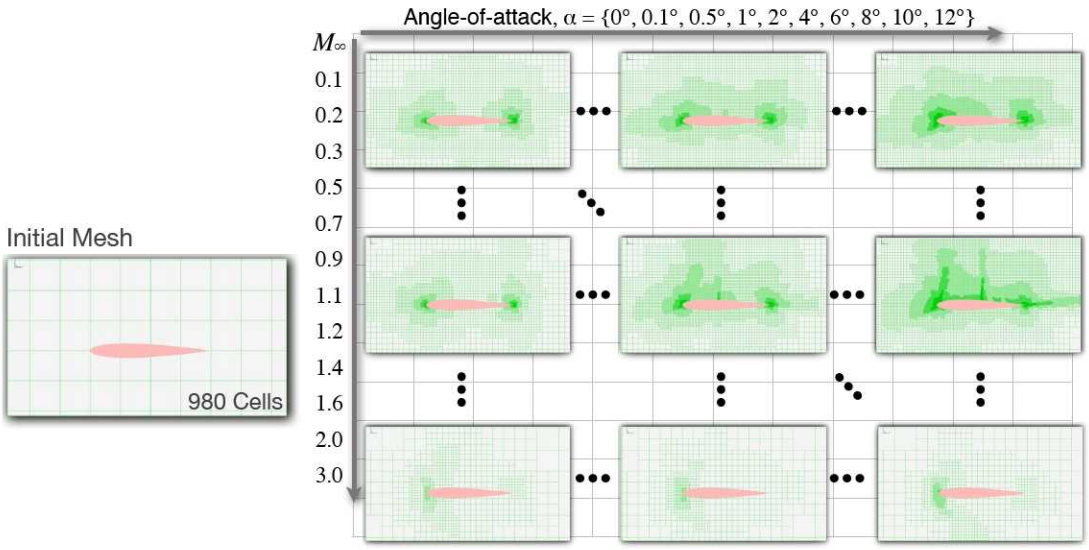

Figure 3: Sample computational meshes in the Mach-α database for NACA 0012 airfoil produced by adjoint-based adaptive mesh refinement to control discretization level in functional $J = C_l + Cd$. Initial mesh with 980 cells shown on left. Final meshes had from 10^3-10^5 cells.

Aerodynamic Coefficients

To control discretization-error to a fixed value, every cases in the database was run adaptively until the estimate of the remaining-error term in eq.(3) was less than 0.008. With the functional in eq.(4), this tolerance translates to 80 counts of drag at zero-lift. Figure 3 shows samples of the final meshes required to achieve this tolerance at various points in the database. This error tolerance was purposely chosen relatively loose since we hope to be able to achieve it over a wide range of flow conditions.

Figure 4 shows simulation results for this database through plots of lift, drag and moment versus alpha for each Mach number. Reassuringly, these data collapse near the incompressible limit, and shock-free inviscid flow shows zero drag for all α. More generally, C_l and C_m behave linearly where expected, while C_d seems to behave nearly quadratically. This presentation, however, actually masks the major sensitivities. To see this more clearly, figure 5 shows performance landscapes of C_l and C_d now plotted as a function of both α *and* Mach number. While the variation with angle-of-attack is generally linear or quadratic, variation with Mach number is much more radical. In this view, the steepness of the carpets imply rapid change in the coefficient with either

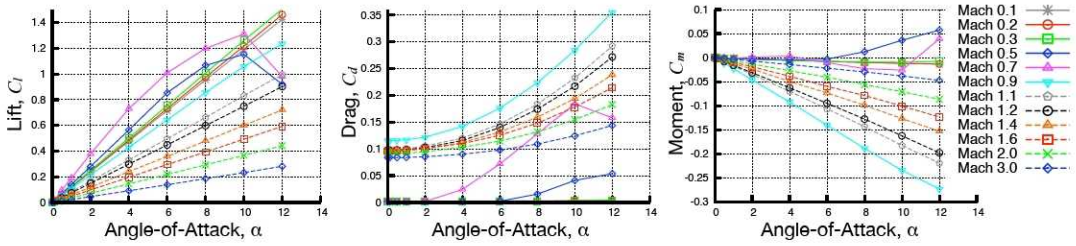

Figure 4: Lift, drag, and quarter-chord pitch-moment for aero-database in figures 2 & 3. All cases computed to constant-error tolerance of 0.008 on the approximation of the remaining-error in the functional $J = C_l + C_d$.

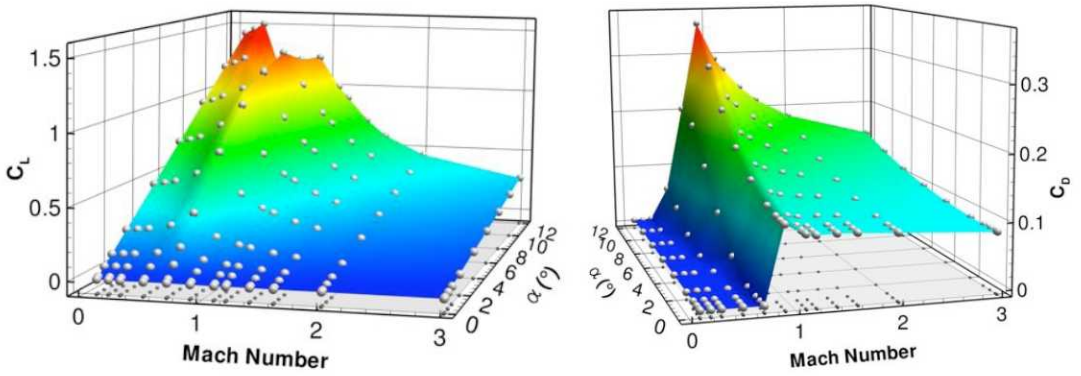

Figure 5: Performance landscapes of C_l & C_d. Steepness of the carpet indicates output sensitivity. All cases computed to constant-error tolerance of 0.008 on the approximation of the remaining-error in the functional $J = C_l + C_d$.

Mach number or α. The response in either output (C_l or C_d) to changes in the flow (Mach or alpha) is precisely the definition of output sensitivity.

Discretization Error and Resolution Requirements

While figures 4 & 5 show the primary aerodynamic data, figure 6 contains the central results for our current investigation of the error landscape. This figure shows quantitative measures of solution quality and cost. At the left of figure 6, we plot the magnitude of the remaining-error term in eq.(3) at each Mach number in the database as a function of angle-of-attack. The plot at the right shows the number of cells required to achieve this error level also as a function of M_∞ and α.

Figure 6 shows the level of remaining error in all the simulations in the database. Our error tolerance of 0.008 appears as a shaded (yellow) plane in Mach-α space. In examining these data we note that all but one case ($M_\infty = 0.5$, $\alpha = 12°$) in the database met the error tolerance. Moreover, we see that the adaptive scheme controlled discretization-error so that the carpet of remaining error forms a relatively flat surface clustered in a narrow band from 0.005 to 0.008. The flatness of this surface is a measure of control.

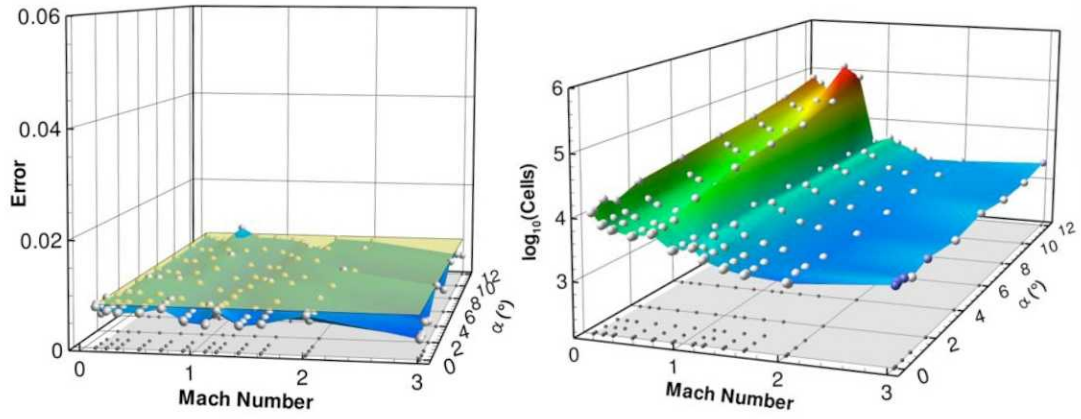

Figure 6: (Left) Remaining-error and (right) number-of-cells carpets for constant-error aero-database. Shaded plane (yellow) on left shows the error tolerance of 0.008 on functional $J = C_l + C_d$. All but one case achieved the tolerance on remaining-error.

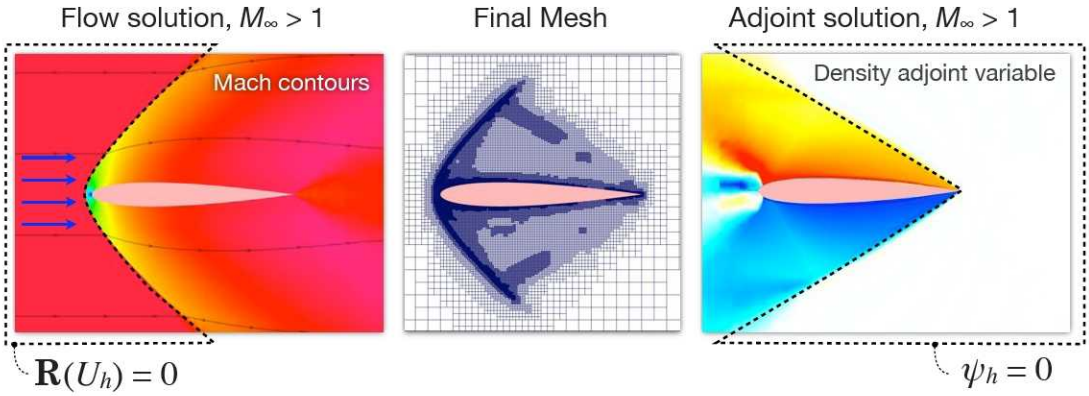

Figure 7: Confinement of domain in supersonic regime. In supersonic flow, mesh adaptation is confined to regions behind the bow shock and in front of the limiting characteristic since the remaining error term in eq.(3) is zero ahead of the bow shock ($\mathbf{R}(U_h) = 0$) and downstream of limiting characteristic, the output functional has no sensitivity to changes in the flow ($\psi_h = 0$).

The carpet plot at the right of figure 6 shows the cell-counts used to achieve this level of error. This plot shows a clear division between the subsonic and supersonic cases which appears as a sharp step-down in the cell-count carpet located just before Mach 1. The supersonic cases generally achieved the desired level of error with over an order of magnitude fewer cells than the subsonic cases. In the subsonic regime, the resolution requirements increase with angle-of-attack and we see increasing Mach number dependencies approaching the incompressible limit and through the transonic regime.

Figure 7 addresses the disparity in cell-counts between the subsonic and supersonic cases. The elliptic nature of subsonic flow means that every point in the domain has the potential to influence in our surface functional $J = C_l + C_d$. Since the entire domain plays a role, mesh adaptation can extend far upstream or downstream. In supersonic flow, however, the domain becomes sharply confined. Upstream of the bow shock, $U_h = U_\infty$ and since U_∞ satisfies the governing equations, $\mathbf{R}(U_h) = 0$. Similarly, downstream of the limiting characteristic, changes in the flow field cannot effect quantities on the body's surface and thus the adjoint variable is zero. Since the remaining error term in eq.(3) is formed by an inner product between the adjoint variables and the primal residuals, these zeros confine mesh enrichment to a small region of the domain between the bow shock and the limiting characteristic. Similar observations were made in ref [14] when discussing figures 9 and 24. This confinement dramatically limits the problem size in supersonic flow. Moreover, as the Mach number increases, this "refinement diamond" contracts in the cross-flow direction and the cell count drops with the area change (volume change in three-dimensions).

In the transonic regime, resolution requirements are driven by the functional's dependence on the precise location of the upper surface shock. Since the shock's position is only known to within a cell, accurate shock placement demands fine mesh cells. As a result, the simulations at Mach 0.7 are among the most cell-intensive in the database. By Mach 0.9 the shocks are attached to the trailing-edge, shock location is unambiguous, and these cases are easy by comparison.

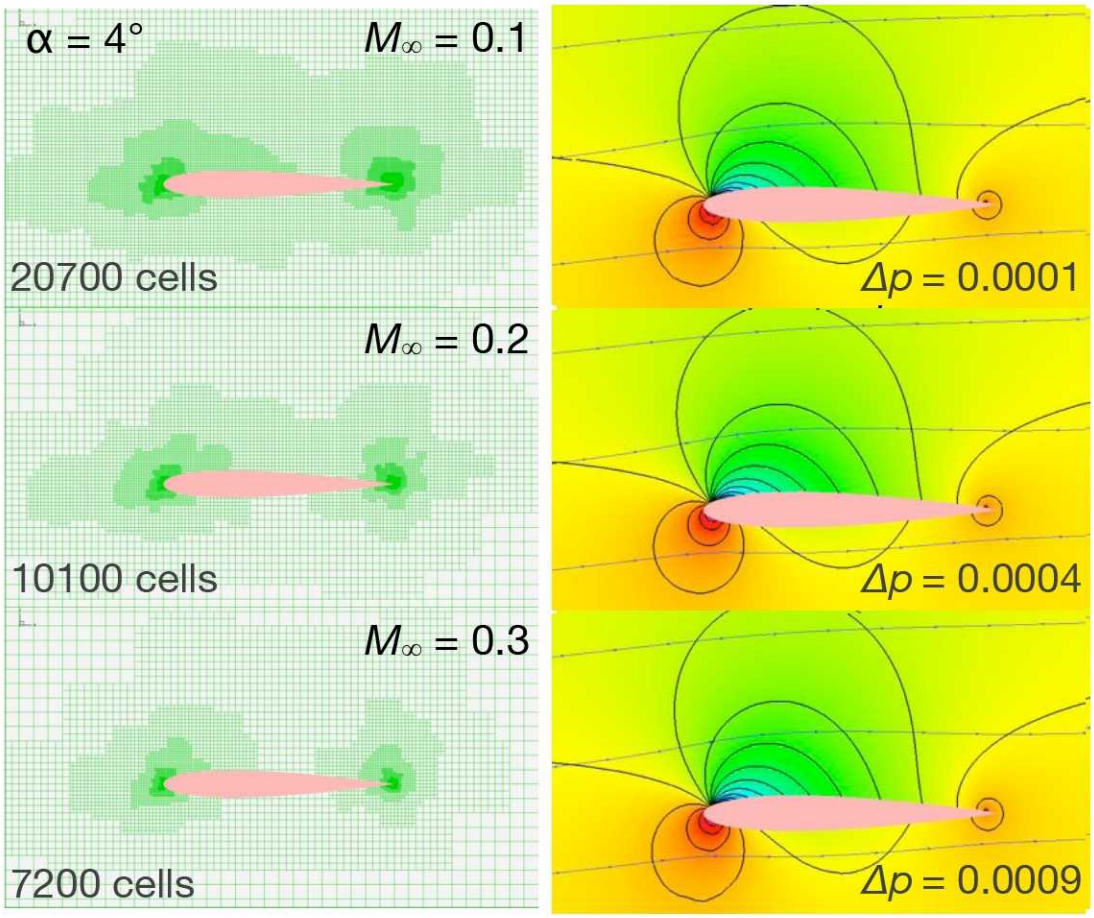

Figure 8: Final meshes and isobars in discrete solutions required to achieve an error-tolerance of 0.008 on functional $J = C_L + C_D$ in near-incompressible flow. $M_\infty = \{0.1, 0.2, 0.3\}$ and $\alpha = 4°$.

While cell-counts in the trans- and supersonic regimes are driven by flow physics, near the incompressible limit they are driven by our model of the governing equations. These simulations were preformed using the compressible form of the governing equations without any low-Mach preconditioning. Near the incompressible limit, the pressure coefficient becomes independent of Mach number giving rise to the well-known $1/M^2$ incompressible scaling. Figure 8 shows isobars in the discrete solution for the three lowest Mach numbers. Isobar levels at M_∞ of 0.1, 0.2 and 0.3 have been chosen to illustrate the approach to self-similarity in these nearly incompressible flows. Pressure signals at $M_\infty = 0.1$ are 9 times weaker than their counterparts at $M_\infty = 0.3$ in accordance with the inverse Mach-squared scaling. This represents nearly a 10 fold decrease in signal strength and therefore meeting the same error-tolerance on the aerodynamic coefficients requires correspondingly higher mesh resolution. This stiffness is simply an artifact of solving the un-preconditioned governing equations and the fact that our functional is based upon coefficients which depend on q_∞.

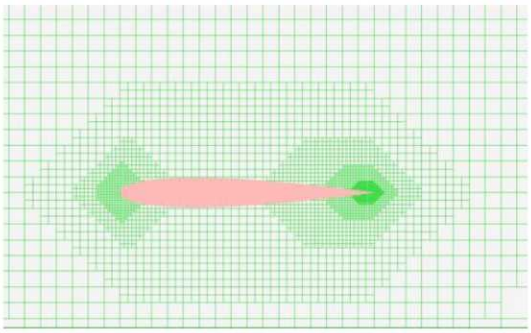

Figure 9: Computational mesh with ~7000 cells used for all cases in the *fixed-mesh* database.

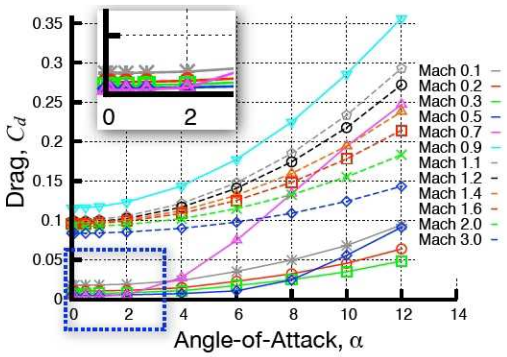

Figure 10: Drag in database computed on *fix mesh*. Inset shows poor prediction of drag at Mach near α = 0.

CONSTANT MESH DATABASE

In contrast to the adaptively refined meshes in the preceding section, we now examine a database computed on a fixed computational mesh. Figure 9 shows a mesh with ~7000 control volumes. This mesh is similar in cell-density and cell-count to some of the adaptive meshes found at moderate subsonic Mach numbers (see fig 3.) and is representative of a "best practices" mesh for this airfoil. Using the same mesh for all simulations in a database is common practice in production CFD making it important to understand the implications of this approach in terms of data quality and discretization error. This fixed-mesh database used the same computational mesh for all 120 simulations in the database. Figure 10 shows the prediction of drag across the entire database as a function of angle-of-attack for all Mach numbers. In comparing these results with those in Fig. 4 we notice immediately the poor prediction in the subsonic regime at low incidence angles. Spurious drag in this region increases as the flow becomes more incompressible, and increases with α. Given the earlier discussion of flow sensitivity in this regime, these results are hardly surprising, but the quantification is illustrative. In shock-free flow, inviscid flow theory predicts zero drag for this airfoil. Instead, Fig.10 shows finite drag for all these cases – with results nicely sorted with the

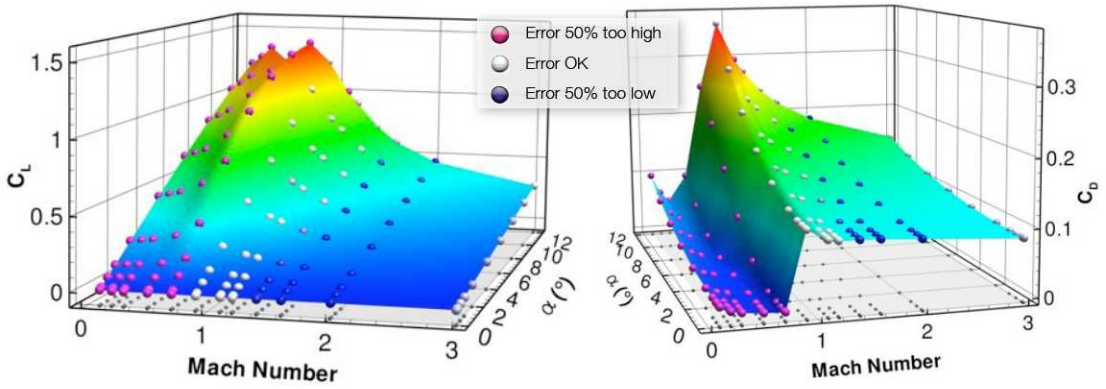

Figure 11: Performance landscapes of C_l & C_d for database computed on a fixed mesh with ~7000 control volumes. Note poor prediction of drag in subsonic shock-free flow. Symbol color indicates level of remaining-error in simulations relative to tolerance of 0.008 on on functional $J = C_L + C_D$.

inverse of Mach number, as we approach the incompressible limit. Figure 11 shows performance landscapes with C_l and C_d plotted as functions of Mach and angle-of-attack. Its interesting to contrast these with their counterparts in fig. 5 and the poor prediction of drag in subsonic shock-free flow is obvious. Symbols on the plot in figure 5 are colored by the value of remaining-error in the simulations. Red symbols mark cases with error at least 50% in excess of our 0.008 tolerance on $J = C_l + C_d$, and virtually all of the subsonic cases are red.

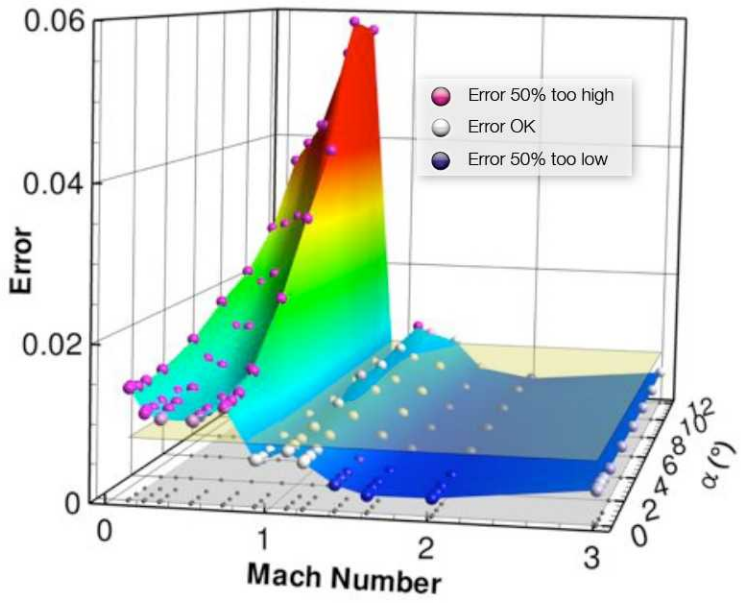

Figure 12: Remaining error landscape of fixed-mesh database. Yellow plane indicates error tolerance of 0.008 in functional $J = C_L + C_D$.

Figure 12 gives a more quantitative understanding of the error-landscape for the constant-mesh database. To facilitate comparison, these data are plotted using the same scale as the error-carpet in the error-controlled database (fig. 6). The yellow plane in fig. 12 shows the tolerance of 0.008 on J. The undersolved simulations (red symbols) are above this plane, while the oversolved ones (blue symbols) lie below. White symbols designate cases that lie within 50% of our desired error tolerance. In general the cases meet or exceed the error tolerance at Mach 0.9 and above. Although all cases were computed on the same mesh, the remaining-error data in fig. 12 vary from 0.016 to 0.06. This means that there is a factor of 40 difference in accuracy of our lift and drag functional between the best and worst cases in the database. In general we observe that the subsonic cases have 5-10 times more error than the supersonic cases, and that the errors are worse in regions of the domain where we the governing equations are stiff (transonic regime and near the incompressible limit).

This last observation underscores the duality between meshing requirements and discretization error. Figure 13 clarifies further. On the left is the "meshing requirements" carpet plot from the error-controlled database replotted from figure 5. On the right, we replot the error carpet from the fixed-mesh database in figure 12, only this time, using a logarithmic scale on the vertical axis. Comparing the two reveals remarkable similarities. We see that the cases with the highest error on the fixed mesh are exactly the same as those which the adjoint has identified as having the most stringent meshing requirements. Cases with high sensitivity have both high meshing requirements and high-error when the fixed-mesh is used for their simulations. Of course, this is not surprising, but we can now quantify the error range and realize that even for a

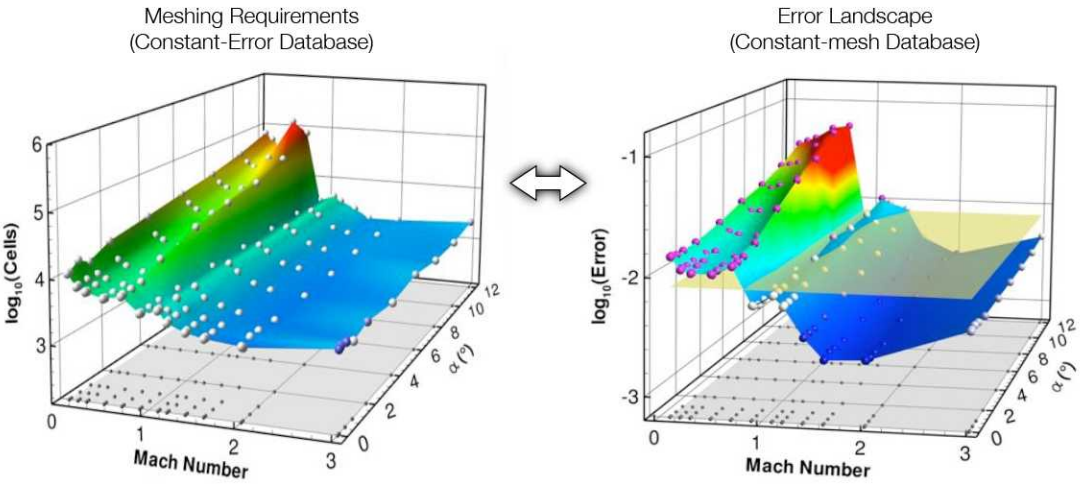

Figure 13: Duality between meshing requirements in constant-mesh database and remaining error landscape of constant-mesh database ($\log_{10}$ scale).

simple 2D airfoil, using a fixed mesh results in output error that spans nearly 2 orders of magnitude. Conversely, attempting to control this error, even in 2D, requires meshes with with cell-counts that vary over two orders of magnitude (1600-200000 cells).

As discussed earlier, cell-counts near the incompressible limit are driven by the inverse Mach-squared incompressible scaling. Cell-counts in the transonic range are driven by the need to resolve the precise location of the shocks, and when this position is only known to within one cell, *h*-refinement is the only meaningful way of improving it. In the high-Mach cases, the domain is substantially confined dramatically reducing meshing requirements, but a certain number of cells are still required to resolve the shock and flowfield behind the shock.

ALTERNATIVE STRATEGIES

In many ways, blindly insisting that all cases in a database achieve a constant error is hardly better than blindly applying the same mesh to all cases. Given the range of cell-counts required to achieve constant error, the hardest few percent of cases will completely dominate the computational cost of the database. In all likelihood a database which truly has uniform error may not be demanded for engineering use. Trajectory simulations, for example have their own set of sensitivities. Cases where the vehicle is likely to spend the majority of its time generally require tighter error tolerances than corners of the flight envelope that may never be reached. Under these circumstances, simply capping the maximum number of cells may be a useful technique for avoiding the needless expense chasing precision in corners of the flight envelope which are unimportant in trajectory simulations or for Guidance and Control system development. On the other hand, its also interesting to consider a relative tolerance where the functional's value is linked to its error tolerance.

CONCLUSIONS AND FUTURE WORK

In order to examine the level of discretization error in simulation-based aerodynamic data, this work compared error levels in two aerodynamic databases using a parallel, multi-level Euler solver on embedded-boundary Cartesian meshes. Discretization errors in user-selected outputs were estimated using the method of adjoint-weighted residuals and adaptive mesh refinement was employed to reduce these errors to specified tolerances. Using this framework, the investigations examined the behavior of discretization error throughout a token database computed for a NACA 0012 airfoil consisting of 120 cases. The accuracy and computational expense were compared for two approaches for aerodynamic database generation. The first approach used mesh adaptation to produce an "error controlled" database where all cases in the database achieved a prescribed level of accuracy. The second approach conducted all simulations using the same computational mesh without adaptation. Error landscapes in both approaches were quantitatively assessed along with the computational costs of both approaches.

This investigation highlighted the interplay between flow field sensitivity, discretization error and meshing requirements. Aerodynamic coefficients in the error-controlled databases demonstrated textbook behavior. However, it was also shown that due to varying flow sensitivity, maintaining a constant error tolerance can require widely disparate mesh sizes. The presence of transonic shocks or the stiffness in the governing equations near the incompressible limit were shown to dramatically increase discretization error requiring additional mesh resolution to control. Results show that such pathologies lead to error levels that vary by over factor of 40 when using a fixed mesh throughout the database. Alternatively, controlling this sensitivity through mesh adaptation leads to mesh sizes which span two orders of magnitude.

We intend to repeat this study in 3D to understand and quantify the impact of dimensionality on both error levels and meshing requirements. Furthermore we intend to examine the usefulness of capping cell-counts to control database cost and examine the utility of relative error-tolerances as a middle-ground between accuracy and expense. Clearly quantitative error estimates on simulation data are of great value and error-estimates are reasonable expectation using techniques that are already available. The dual nature of meshing requirements and discretization error implies that even in the absence of mesh adaptation, the error estimates offered by the method of adjoint-weighted residuals adjoint provides insight into both flow sensitivity and meshing requirements.

REFERENCES

1. TOP500 List Highlights, Nov. 2008. *http://www.top500.org/lists/2008/11/highlights*
2. Biswas, R., Aftosmis, M.J., Kiris, C., and Shen, B.-W., "2007: Petascale Computing: Impact on Future NASA Missions." In Petascale Computing: Algorithms and Applications (D. Bader, ed.), Chapman and Hall /CRC Press, Dec. 2007.

3. Mavriplis, D.J., Aftosmis, M.J., and Berger, M.J., "High resolution aerospace applications using the NASA Columbia supercomputer." *International Journal of High Performance Computing Applications*. **21**(1):106-126. Jan. 2007
4. Pulliam, T.H., and Jespersen, D.C., "Large Scale Aerodynamic Calculation on Pleiades", 21st Internat. Conf. on Parallel CFD, Moffett Field CA., May 2009.
5. Aftosmis, M.J., Berger, M.J., Melton, J.E., "Robust and Efficient Cartesian Mesh Generation for Component-Based Geometry."*AIAA J.* **36**(6):952-960, Jun. 1998.
6. Aftosmis, M.J., Berger, M.J., and Adomavicius, G., "A parallel multilevel method for adaptively refined Cartesian grids with embedded boundaries." *AIAA Paper 2000-0808*, Jan. 2000.
7. Aftosmis, M.J., Berger, M.J, Biswas, R., Djomehri, M.J., Hood, R., Jin, H., and Kiris, C., "A Detailed Performance Characterization of Columbia using Aeronautics Benchmarks and Applications," *AIAA Paper 2006-0084*, Jan. 2006.
8. Rogers, S. E, Aftosmis, M.J., Pandya, S.A., and Chaderjian, N.M, Tejnil, E., and Ahmad, J., "Automated CFD parameter studies on distributed parallel computers." *AIAA Paper 2003-4229*. 16th AIAA Computational Fluid Dynamics Conference, Jun. 2003.
9. Chaderjian, N.M., Rogers, S.E., Aftosmis, M.J., Pandya, S.A., Ahmad, J.U., and Tejnil, E., "Automated CFD Database Generation for a 2nd Generation Glide-Back-Booster," *AIAA Paper 2003-3788*, June 2003.
10. Murman, S.M., Aftosmis, M.J., and Nemec, M., "Automated parameter studies using a Cartesian method." *AIAA Paper 2004-5076*, Aug. 2004.
11. Aftosmis, M.J., and Rogers S. E., "Effects of jet-interaction on pitch control of a launch abort vehicle," *AIAA Paper 2008-1281*, Jan. 2008.
12. Nemec, M., and Aftosmis, M.J., "Adjoint error-estimation and adaptive refinement for embedded-boundary Cartesian meshes." *AIAA Paper 2007-4187*, 18th AIAA CFD Conference, Miami, FL., Jun. 2007.
13. Nemec, M., and Aftosmis, M.J., "Adjoint Sensitivity Computations for an Embedded-Boundary Cartesian Mesh Method," Jol. Computational Physics (2007), doi:10.10.16/j.jcp.2007.11.018. *http://dx.doi.org/10.1016/j.jcp.2007.11.018* (persistent link)
14. Nemec, M., Aftosmis, M.J., and Wintzer M., "Adjoint-based adaptive mesh refinement for complex geometries," *AIAA Paper 2008-0725*, Jan. 2008.
15. Nemec, M., and Aftosmis, M.J., "Adjoint Error Estimation and Adaptive Refinement for Embedded-Boundary Cartesian Meshes." *AIAA J.* to Appear, 2009.
17. Venditti, D. A. and Darmofal, D. L., "Grid Adaptation for Functional Outputs: Application to Two-Dimensional Inviscid Flow," *Journal of Computational Physics* , (176):40-69, 2002.